

21

in the case of a memory device (block 1404. These device parameters may be determined on-the-fly, as well as by the manufacturer of the device. Next, the process 1400 determines if it should obtain a corresponding graphics object. If not, it proceeds to process block 1418, where it displays the device parameter information, and exits. Otherwise, it obtains the corresponding graphics object name (process block 1408). The process 1400 then advances to process block 1410, where it creates a corresponding data packet for the Attribute Generator. The data packet is then used to invoke the Attribute Generator (process block 1412). Once invoked, the Attribute Generator issues a command to the Image Generator to display the corresponding image (process block 1414). The device information obtained earlier (e.g., the device parameters) is displayed, and the Image Generator displays the corresponding graphics (process block 1416). The process 1400 then exits.

FIG. 15 is a flowchart illustrating one embodiment of a processor information and image handler process 1500 provided in accordance with the principles of the invention. Beginning from a start state, the process 1500 obtains the processor vendor identification code from firmware (process block 1502). Next, the processor operating speed device parameter is obtained (process block 1504). In one embodiment, this includes a value representative of the manufacturer-determined operating speed, and a value representative of the operating speed measured on-the-fly. Next, the graphic object name for the corresponding processor is obtained from a processor mapping table (process block 1506). The process 1500 then creates a corresponding data packet for the Attribute Generator (process block 1508). The Attribute Generator is invoked with the data packet (process block 1510), which in turn invokes the Image Generator (process block 1512). The processor information (both manufacturer determined device parameters and those measured on-the-fly) is displayed, along with the corresponding image. The process 1500 then exits.

Thus, the present invention is an efficient technique to generate a media output. The technique generates a graphic attribute characterizing the graphic object from a template. The image is then generated from the graphic object using the graphic attribute. In another embodiment, a media device such as a graphics controller or an audio card is initialized during the early BIOS boot-up. Then a media operation is performed while the normal BIOS tasks are being performed. The present invention also provides system device information such as values representative of device parameters, that is generated on-the-fly, and also displays such information simply and efficiently.

The present invention also provides an efficient technique to generate a media output. The media output includes system device information (e.g., device parameters) that is generated on-the-fly, and also displays such information simply and efficiently. The technique generates a graphic attribute characterizing the graphic object from a template. The image is then generated from the graphic object using the graphic attribute. In another embodiment, a media device such as a graphics controller or an audio card is initialized during the early BIOS boot-up. Then a media operation is performed while the normal BIOS tasks are being performed.

The present invention also provides a method and apparatus to provide functionalities to system BIOS in a simple, elegant and flexible manner. An interface module is provided, which enables various parties such as PC i system manufacturers and motherboard manufacturers to provide additional system BIOS functionalities with minimal impact

22

to the system BIOS code. The method comprises interfacing an interface module to the system BIOS, and receiving a request from the system BIOS to perform a task. System device information such as values representative of device parameters, associated with the task is received from the system BIOS. The interface module translates the system device information to provide translated information. The translated information is then transferred to a corresponding module.

While this invention has been described with reference to illustrative embodiments, this description is not intended to be construed in a limiting sense. Various modifications of the illustrative embodiments, as well as other embodiments of the invention, which are apparent to persons skilled in the art to which the invention pertains are deemed to lie within the spirit and scope of the invention.

What is claimed is:

1. A method to display information comprising:

retrieving, during a boot period, a first value representative of a first device parameter;

detecting, during the boot period, a second value representative of a second device parameter;

retrieving a template corresponding to a graphic object from a storage, said template including a plurality of commands;

generating a graphic attribute based on one or more of said commands from the template, the graphic attribute characterizing the graphic object;

displaying said first and second values; and

generating an image, during a boot period of the computer system, from the graphic object according to the graphic attribute.

2. The method of claim 1, wherein the first device parameter is a parameter determined at the time of manufacture.

3. The method of claim 2, wherein the second device parameter is a parameter determined on-the-fly.

4. The method of claim 3, wherein the first and second device parameters are processor operating speeds.

5. The method of claim 1 wherein the graphic attribute includes one of a location of the graphic object, a position of display, a size, a zone, a graphic primitive set, an animation primitive set, and a sound set.

6. The method of claim 5, wherein the graphic object is one of a banner, a logo, an advertisement item, and a textual item.

7. The method of claim 5 further comprising:

initializing a graphic controller to set a display environment; and

writing the image to a display buffer associated with the graphic controller.

8. The method of claim 5, wherein generating the graphic attribute comprises:

scanning the template to generate tokens; and

parsing the tokens to generate the graphic attribute.

9. The method of claim 5, wherein generating the image comprises:

retrieving the graphic object from the storage based on the graphic attribute; and

transforming the graphic object to the image based on the graphic attribute.

10. The method of claim 5 wherein the zone includes one of a graphic zone, a textual zone, and an animation zone.

11. The method of claim 5, wherein the storage is a non-volatile writable memory.